

MULTI-CPU SYSTEM HAVING UPDATE PROCESSING FUNCTION OF FIRMWARE

Publication number: JP10260845

Publication date: 1998-09-29

Inventor: MURAKAMI HIROYUKI

Applicant: FUJITSU LTD

Classification:

- International: G06F15/16; G06F9/445; G06F15/177; G06F15/16;
G06F9/445; (IPC1-7): G06F9/445; G06F15/177

- european:

Application number: JP19970065849 19970319

Priority number(s): JP19970065849 19970319

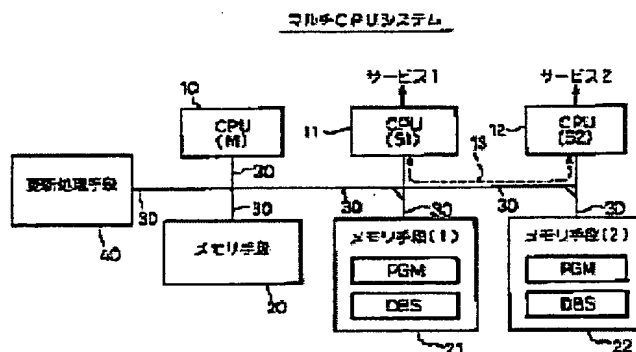
Report a data error here

Abstract of JP10260845

PROBLEM TO BE SOLVED: To highly efficiently execute a software down loading processing while suppressing the influence to regular service to the minimum, by permitting a master CPU to directly rewrite a program to a memory means corresponding to slave CPUs at the time of rewriting the program.

SOLUTION: An update processing means 40 directly writes the program for the memory means 21 and 22 corresponding to slave CPUs 11 and 12 being the objects of update without interfering mutual communication 13 between the slave CPU 11 and 12 and the execution of service at the time of rewriting at least the program for updating firmware.

Namely, the respective processings of software down loading and program switching are integrated into the master CPU. Consequently, contention with regular service in the use of the CPU resource can be avoided.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-260845

(43) 公開日 平成10年(1998) 9月29日

(51) Int.Cl.⁶

G 0 6 F 9/445

15/177

識別記号

F I

G 0 6 F 9/06

15/16

4 2 0 M

4 2 0 S

審査請求 未請求 請求項の数13 O L (全 13 頁)

(21) 出願番号

特願平9-65849

(22) 出願日

平成9年(1997) 3月19日

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番
1号

(72) 発明者 村上 寛之

神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(74) 代理人 弁理士 石田 敬 (外3名)

(54) 【発明の名称】 ファームウェアの更新処理機能を有するマルチCPUシステム

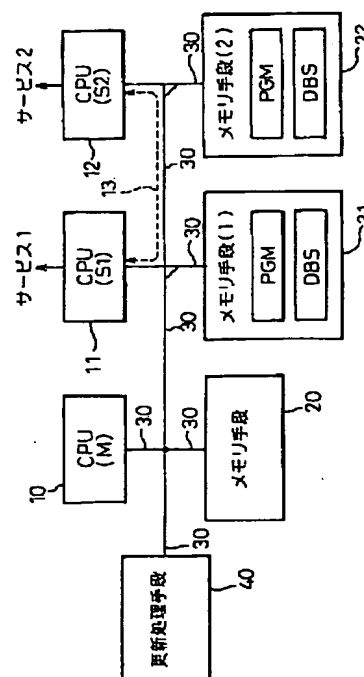
(57) 【要約】

【課題】 複数のCPUを有するマルチCPUシステムにおいて、各CPUに対するソフトウェアダウンロード処理を、他のCPUが実行中の通常サービスに殆んど影響を与えずに行えるようにする。

【解決手段】 マスターCPU10と、複数のスレーブCPU(11, 12)と、各スレーブCPU(11, 12)によってアクセスされる固有のプログラムやデータベースを格納するメモリ手段(21, 22)と、このメモリ手段(21, 22)の各々とマスターCPU10とを直接接続する接続手段30と、プログラムおよびデータベースを書き換える際、マスターCPU10が接続手段40を介して直接、メモリ手段(21, 22)に対し、当該プログラムやデータベースの書き込みを行う更新処理手段40とからなる。

本発明の基本構成を示す図

マルチCPUシステム



【特許請求の範囲】

【請求項1】 マスターCPUと、

複数のスレーブCPUと、

該複数のスレーブCPUの各々に対応して設けられ、各該スレーブCPUに固有のサービスを実行するために、各該スレーブCPUによってアクセスされる固有のプログラムを少なくとも格納する複数のメモリ手段と、前記複数のメモリ手段の各々と前記マスターCPUとを直接接続する接続手段と、

ファームウェアの更新のために少なくとも前記プログラムを書き換える際、前記複数のスレーブCPU相互のCPU間通信ならびに前記サービスの実行に干渉することなく、前記マスターCPUが前記接続手段を介して直接、該更新の対象となる前記スレーブCPUに対応する前記メモリ手段に対し、当該プログラムの書込みを行う更新処理手段とからなることを特徴とするファームウェアの更新処理機能を有するマルチCPUシステム。

【請求項2】 各前記メモリ手段は、前記プログラムを格納するプログラムメモリと、前記サービスに関連するデータを格納するデータベースとを有する請求項1に記載のマルチCPUシステム。

【請求項3】 各前記メモリ手段は前記プログラムを格納するプログラムメモリを含んでなり、該プログラムメモリは、

該メモリ手段に対応する前記スレーブCPUが前記サービスの実行のために直接アクセスする現用面と、前記マスターCPUが前記ファームウェアの更新のために直接アクセスする予備面との2面構成からなる請求項1に記載のマルチCPUシステム。

【請求項4】 各前記メモリ手段は前記サービスに関連するデータを格納するデータベースを含んでなり、該データベースは、

該メモリ手段に対応する前記スレーブCPUが前記サービスの実行のために直接アクセスする現用面と、前記マスターCPUが前記ファームウェアの更新のために直接アクセスする予備面との2面構成からなる請求項1に記載のマルチCPUシステム。

【請求項5】 各前記メモリ手段は、前記プログラムを格納するプログラムメモリを有し、かつ、

該プログラムメモリは、該メモリ手段に対応する前記スレーブCPUが前記サービスの実行のために直接アクセスする現用面と、前記マスターCPUが前記ファームウェアの更新のために直接アクセスする予備面との2面構成からなり、

前記更新処理手段は、前記マスターCPUの指示により、前記ファームウェアの更新に際し、前記プログラムメモリにおける前記現用面と前記予備面との間の切替えを実行する請求項1に記載のマルチCPUシステム。

【請求項6】 各前記メモリ手段は、前記サービスに関連するデータを格納するデータベースを有し、かつ、

前記データベースは、前記メモリ手段に対応する前記スレーブCPUが、前記サービスの実行のために直接アクセスする現用面と、前記マスターCPUが前記ファームウェアの更新のために直接アクセスする予備面との2面構成からなり、

前記更新処理手段は、前記マスターCPUの指示により、前記ファームウェアの更新に際し、前記データベースにおける前記現用面と前記予備面との間の切替えを実行する請求項1に記載のマルチCPUシステム。

10 【請求項7】 前記更新処理手段は、前記マスターCPUの指示により、前記プログラムメモリにおける前記現用面と前記予備面との間の切替えを実行した直後に、該プログラムメモリに対応する前記スレーブCPUをリセットして該スレーブCPUをリスタートさせる請求項5に記載のマルチCPUシステム。

【請求項8】 前記更新処理手段は、前記マスターCPUの指示により、前記データベースにおける前記現用面と前記予備面との間の切替えを実行した直後に、該データベースに対応する前記スレーブCPUをリセットして該スレーブCPUをリスタートさせる請求項5に記載のマルチCPUシステム。

【請求項9】 前記更新処理手段は、前記スレーブCPUをリスタートさせた後、該スレーブCPUが正常に立ち上がったか否かのステータスを監視して、正常に立ち上がっていないことが検知されたとき、前記現用面と前記予備面との間の切替えを切り戻して元の状態を維持する請求項7に記載のマルチCPUシステム。

【請求項10】 各前記メモリ手段は前記プログラムを格納するプログラムメモリを含んでなり、該プログラムメモリは、該メモリ手段に対応する前記スレーブCPUが前記サービスの実行のために直接アクセスする現用面と、前記マスターCPUが前記ファームウェアの更新のために直接アクセスする予備面との2面構成からなると共に、

各前記メモリ手段は前記サービスに関連するデータを格納するデータベースを含んでなり、該データベースは、該メモリ手段に対応する前記スレーブCPUが、前記サービスの実行のために直接アクセスする現用面と、前記マスターCPUが前記ファームウェアの更新のために直接アクセスする予備面との2面構成からなり、

ここに、前記マスターCPUが有するメモリ空間の一部に定義されるペリフェラルI/Oの領域に前記更新処理手段を実現し、

該ペリフェラルI/Oは複数の更新処理レジスタを予め定義していて、該複数の更新処理レジスタは、各前記スレーブCPU毎に、前記プログラムメモリの前記現用面および前記予備面を個別に形成する第1バンクおよび第2バンクがいずれの面として機能しているかを表示する第1レジスタを含む請求項1に記載のマルチCPUシステム。

【請求項11】 前記更新処理レジスタは、各前記スレーブCPU毎に、前記データベースの前記現用面および前記予備面を個別に形成する第1バンクおよび第2バンクがいずれの面として機能しているかを表示する第2レジスタを含む請求項10に記載のマルチCPUシステム。

【請求項12】 前記更新処理レジスタは、前記マスターCPUが前記スレーブCPUに対しリセットを指示する表示を行うために各前記スレーブCPU毎に設けた第3レジスタを含み、該第3レジスタには、前記現用面および前記予備面の間の切替えが実行された直後に当該スレーブCPUをリセットすべきことを指示するための表示を行う請求項10に記載のマルチCPUシステム。

【請求項13】 前記更新処理レジスタは、前記マスターCPUが前記スレーブCPUに対しリセットをかけてこれをリスタートさせたとき正常に立ち上がったか否かを表示するために各前記スレーブCPU毎に設けた第4レジスタを含む請求項12に記載のマルチCPUシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、ファームウェアの更新処理機能を有するマルチCPUシステムに関する。マルチCPU(Central Processing Unit)システムは、複数のCPUと、これらCPUの各々に連係するメモリ手段とを具備してなり、ユーザに対し、多種のサービスを提供する。

【0002】上記のサービスは固定で不変の場合もあるが、一般には、上記ユーザの要求に応じて可変となることが多い。このようなユーザからの要求があったときは、上記マルチCPUシステムにより構成されるファームウェア(firmware)を新たな形態に変更しなければならない。これがファームウェアの更新(update)である。本発明は、かかる、ファームウェアの更新処理機能を有するマルチCPUシステムについて述べる。

【0003】

【従来の技術】上述したマルチCPUシステムは、各種産業分野における各種の装置等に種々の形式で導入されるものであるから、本発明においてはそのマルチCPUシステムが如何なる装置を対象とするかについては特に限定しない。しかし発明の理解のために、対象となる装置の好適例の一つ挙げ、以下、この好適例を参照しながら説明を行うこととする。

【0004】図11は本発明が適用される好適な装置例を説明する図である。本図は代表的なスイッチングネットワークを概略的に示している。スイッチングネットワークは、複数の交換機(図中、EX(exchange)で示す)相互間を、各伝送路Lを介し、接続して構成され、各交換機EXは配下に多数の加入者(電話端末あるいはデータ端末)SUB(subscriber)を収容する。これら

交換機(EX)相互間を各伝送路Lを介して接続するために伝送装置(TR(transmission apparatus))が、各交換機対応に設けられる。伝送装置TRはまた、スイッチングネットワーク全体を監視し制御するための監視センターと各交換機EXとを接続するためにも設けられる。

【0005】このようなスイッチングネットワークの構成要素のうち、本発明は上記伝送装置TRに適用することができる。この伝送装置は、信号の伝送制御機能や回線の状況監視機能等、種々の機能を果すものである。このため、これら種々の機能にそれぞれ対応させて種々のCPUが導入され、全体としてマルチCPUシステムを形成している。

【0006】従来の伝送装置TRにおいては、ユーザからの要求に応じて、所要のプログラムを格納するROMに差し替えることによって、上記ファームウェアの更新を行っていた。しかしながらこの従来の手法では多大な工数と時間を要するため、次のような手法が実用に供されている。これは、上記監視センターから、通信ラインを経由して各伝送装置TRにプログラムファイルを転送し、その転送されたプログラムファイルを、各々がCPUを搭載する各ユニット上のメモリに書き込むことによって、ソフトウェアダウンロードを行い、ファームウェアの更新処理を行うというものである。

【0007】一方、伝送装置TRにおいては装置の複雑化が進み、装置のアーキテクチャはCPUの数が増加する方向にある。したがって、それら複数のCPUに対するソフトウェアダウンロードを行う場合、プログラムファイル数は増加し、そのサイズも増大し、また新旧プログラムの切替え制御も複雑化する傾向にある。かくしてソフトウェアダウンロード処理に要する負荷は増大するが、CPUに与える負荷やCPU間通信に要する負荷は増大させないようにして、通常サービスへの影響を最小限にすることが望まれる。

【0008】従来のソフトウェアダウンロード処理は、各CPUのそれぞれに、ソフトウェアおよびデータベースの書き込み/切替え制御機能を備えることによって実行するため、該ソフトウェアダウンロード処理は、通常サービスを行うアプリケーション処理と共存することになる。すなわち1つのCPU資源を、これら2つの処理によって共用することになる。したがって例えば各CPUへの新旧プログラムの切替え要求や新プログラムによるCPUの立ち上がりの確認等の制御は、CPU間で電文をやりとりすることにより行う。また、この切替え時のデータ転送は、通常サービス処理のために供されるCPU間通信用のバスを共用して行うと共に、そのデータ転送方法と同様に、CPU間通信用の電文形式をとっている。

【0009】

【発明が解決しようとする課題】上述のように従来は、各CPUのそれぞれに、ソフトウェアおよびデータベースの書込みおよび切替え制御機能を持つように構成する。したがって通常サービスを行うアプリケーション処理と、ソフトウェアダウンロード処理とが、同一のCPU資源を共用することになる。このためCPUの処理負荷は高くなり、上記両処理が互いに圧迫し合う結果となる。

【0010】また、ソフトウェアダウンロード処理を各CPUがそれぞれ独自に行うことから、上記アプリケーション処理がもし暴走したとすると、当該CPUにおいては、新たなソフトウェアの書込みおよび新たなソフトウェアの切替え、すなわちファームウェアの更新処理を行うことが不可能になる。また、新たなプログラムへの切替え制御を実行するための、CPU間における電文のデータ転送について見ると、通常サービスのためのデータ転送と同一のバスを用いた、CPU間のシリアル通信であることから、大量のプログラムデータや大量のデータベースファイルのデータを転送することによって通信負荷は極端に増大してしまう。また、このために高速のデータ転送もきわめて困難になる。

【0011】したがって本発明は、各CPUに対するソフトウェアダウンロード処理、すなわちファームウェアの更新処理を、通常サービスへの影響を最小限に抑えつつ、高効率で行うことのできるマルチCPUシステムを提供することを目的とするものである。

【0012】

【課題を解決するための手段】図1は本発明の基本構成を示す図である。本図に示すとおり、本発明に係るマルチCPUシステムは、マスターCPU(M)10と、複数の、例えば2台の、スレーブCPU(S1)11およびスレーブCPU(S2)12と、複数のメモリ手段21および22等と、接続手段30と、更新処理手段40とからなる。

【0013】複数のメモリ手段21および22は、複数のスレーブCPU11および12の各々に対応して設けられ、各スレーブCPU11および12に固有のサービス(“サービス1”および“サービス2”)を実行するために、各スレーブCPU11および12によってアクセスされる固有のプログラムを少なくとも格納する。接続手段30は、複数のメモリ手段21および22の各々とマスターCPU10とを直接接続する。

【0014】更新処理手段40は、ファームウェアの更新のために少なくともプログラムを書き換える際、複数のスレーブCPU(11, 12)相互のCPU間通信13ならびに上記のサービスの実行に干渉することなく、マスターCPU10が接続手段30を介して直接、その更新の対象となるスレーブCPU(11, 12)に対応するメモリ手段(21, 22)に対し、当該プログラムの書込みを行う。

【0015】また、メモリ手段21および22の各々は、上記のプログラムを格納するプログラムメモリPGMと、上記のサービスに関連するデータを格納するデータベースDBSとを有する。さらにまた、図1には記載していないが、後述するように、プログラムメモリPGMは、メモリ手段(21, 22)に対応するスレーブCPU(11, 12)が、上記のサービスの実行のために直接アクセスする現用面と、マスターCPU10が、上記のファームウェアの更新のために直接アクセスする予備面との2面構成からなる。

【0016】同様に、データベースDBSは、メモリ手段(21, 22)に対応するスレーブCPU(11, 12)が、上記サービスの実行のために直接アクセスする現用面と、マスターCPU10が、上記のファームウェアの更新のために直接アクセスする予備面との2面構成からなる。

【0017】図1の基本構成をとることにより下記i)~v)の特徴が生ずる。i) スレーブCPU11および12のメモリ手段21および22へプログラムを書き込みまたおよびデータベースデータを書き込むに際し、マスターCPU10と、スレーブCPU11, 12のメモリ手段21, 22とが直接、接続手段30により接続されたアーキテクチャをとり、CPU間通信13を介さずにマスターCPU10が直接メモリ手段(PGM)へプログラムを書き込み可能なソフトウェアダウンロードが実現される。ii) マスターCPU10が、後述するペリフェラルI/Oを制御することにより、スレーブCPU側メモリ手段の前記現用面/予備面の切替えや切替時におけるスレーブCPUへのリスタート指示を行い、新旧プログラムの切替え動作をマスターCPU10が直接制御することが可能となる。iii) マスターCPU10が、ソフトウェアダウンロード処理とこれに続く新旧プログラムの切替え制御とを一括して集中的に行うことにより、アプリケーション処理と、ソフトウェアダウンロードと新旧プログラム切替え処理との間に独立性を保つことができ、ソフトウェアダウンロードと新旧プログラム切替え処理と、アプリケーション処理とを並行して行うことができる。iv) スレーブCPU(11, 12)のメモリ手段(21, 22)へのプログラムやデータベースデータの書込み制御と、新旧プログラムの切替え制御とを、マスターCPU10が一括して行うため、デバイス異常等に起因したハードウェア障害や、あるいはソフトウェア障害によって上記現用面のプログラムが正常に動作しなくなっても、ソフトウェアダウンロードおよび新旧プログラム切替えは、その障害の発生に拘らず実行可能となる。v) スレーブCPU(11, 12)のメモリ領域(PGM)にプログラムの書込みが行われていない状態でも、マスターCPU10により当該スレーブCPUに対して、ソフトウェアダウンロードや新旧プログラム切替えは実行可能とする。

【0018】

【発明の実施の形態】図2は本発明に係るマルチCPUシステムの一実例を示す図(その1)、図3は同図(その2)である。なお、全図を通じて同様の構成要素には同一の参照番号または記号を付して示す。図2において、24はマスタープログラムメモリ(M・PGM)であり、図1のメモリ手段20に相当する。25は第1スレーブプログラムメモリ(S1・PGM)であり、図1のメモリ手段20に相当する。26は第2スレーブプログラムメモリ(S2・PGM)であり、図1のメモリ手段(1)21に相当する。

【0019】さらに27は第1スレーブデータベース

(S1・DBS)であり、図1のメモリ手段(1)21に属し、28は第2スレーブデータベース(S2・DBS)であって、図1のメモリ手段(2)に属する。また図2および図3に示す31はバスであり、図1の接続手段に相当する。この図3に表すブロック41は図1に示す、ファームウェアの更新処理手段30に対応し、本図ではペリフェラルI/O(peripheral I/O)として実現している。

【0020】図2において、メモリ24、25および26はそれぞれ現用面ACT(active)と予備面STBY(standby)とに機能分割されている。同様に、データベース27および28もそれぞれに現用面ACTと予備面STBYとに機能分割されている。ここで注意すべきことは、上記メモリ24、25および26とデータベース27および28を見ると、各々の左側を現用面ACTとし右側を予備面STBYとしているが、これはある時点での分担状況を示したものであって、他の時点ではその分担状況が各メモリ(24~26)毎にまた各データベース(27、28)毎に反転し、左側が予備面STBYとなり右側が現用面ACTになる場合もある。要するに、プログラムメモリ(PGM)とデータベース(DBS)はそれぞれ第1バンク(bank) #1と第2バンク(bank) #2の2面を持ち、一方のバンク(例えば#1)が現用面ACTであるときは他方のバンク(例えば#2)が予備面STBYとなる。

【0021】図3において、ファームウェアの更新処理手段40をなすペリフェラルI/O41は、マスターCPU10に対応する区分Mと、第1スレーブCPU(S1)11に対応する区分S1と、第2スレーブCPU(S2)12に対応する区分S2によって定義される。これらの区分S1およびS2はいずれも同様に細区分され、プログラムメモリ(PGM)の上記バンク(bank)の切替えに関連する小区分と、データベース(DBS)の上記バンク(bank)の切替えに関連する小区分と、それぞれ対応するCPU(CPU(S1)、CPU(S2))に対するリセット(reset)に関連する小区分と、そのCPU(CPU(S1)、CPU(S2))のステータス(status)に関連する小区分とからなる。

【0022】区分Mについてもほぼ同様に、プログラムメモリ(PGM)の上記バンクの切替えに関連する小区分と、そのCPU、すなわちCPU(M)10に対するリセットに関連する小区分と、そのCPU(M)10のステータスに関連する小区分とからなる。本発明の特徴は、上記の図2および図3に一層具体的に説明したように、

i) 各メモリ手段21、22は、プログラムを格納するプログラムメモリ25、26を有し、かつ、プログラムメモリ25、26は、メモリ手段21、22に対応するスレーブCPU11、12がサービスの実行のために直接アクセスする現用面ACTと、マスターCPU10がファームウェアの更新のために直接アクセスする予備面STBYとの2面構成からなる。そして更新処理手段40(41)は、マスターCPU10の指示により、ファームウェアの更新に際し、プログラムメモリ25、26における現用面ACTと予備面STBYとの間の切替えを実行する。

【0023】ii) 各メモリ手段21、22は、サービスに関連するデータを格納するデータベース27、28を有し、かつ、それらデータベース27、28は、メモリ手段21、22に対応するスレーブCPU11、12が、サービスの実行のために直接アクセスする現用面ACTと、マスターCPU10がファームウェアの更新のために直接アクセスする予備面STBYとの2面構成からなる。そして更新処理手段40(41)は、マスターCPU10の指示により、ファームウェアの更新に際し、データベース27、28における現用面ACTと予備面STBYとの間の切替えを実行する。

【0024】iii) 更新処理手段40(41)は、マスターCPU10の指示により、プログラムメモリ25、26における現用面ACTと予備面STBYとの間の切替えを実行した直後に、これらプログラムメモリ25、26に対応するスレーブCPU11、12をリセット(reset)してスレーブCPU11、12をリスタート(res tart)させる。

【0025】iv) 更新処理手段40(41)は、マスターCPU10の指示により、データベース27、28における現用面ACTと予備面STBYとの間の切替えを実行した直後に、これらデータベース27、28に対応するスレーブCPU11、12をリセットしてスレーブCPU11、12をリスタートさせる。図2および図3についてさらに詳細に説明する。

【0026】前述のとおり、“M”はマスターCPU(10)であり、“S1”、“S2”はソフトウェアダウロード対象であるスレーブCPU(11、12)である。S1、S2はプログラム格納用の不揮発領域を2面(バンク#1、バンク#2)持ち、運用状態にあるプログラムが書かれている面がACT-バンクであり、もう一方の面がSTBY-バンクである。S1、S2は、

データベース (DBS) 格納用の不揮発領域を2面 (バンク #1、バンク #2) 持ち、これらも同様にACTーバンク、STBYーバンクの状態をもつ。ペリフェラル I/O 41には、不揮発領域のバンクのACTおよびSTBY状態の切替えを行うバンク切替え設定部分、当該CPUに対しCPUリセットをかけるCPUリセット設定部分、CPUの障害等の状態を示すCPUステータス設定部分が配置されている。

【0027】マスターCPU10と各スレーブCPU11、12の不揮発領域 (プログラムメモリとデータベース) およびペリフェラル I/O 41は、バス31で接続されていて、図4および図5に示すように、これらは全てマスターCPU10が有するメモリ空間 (メモリマップ) 上にマッピングされている。したがってマスターCPU10から容易にアクセス可能である。

【0028】図4はマスターCPUが有するメモリ空間を表す図 (その1)、図5は同図 (その2) である。ただし、マスターCPU10が有するメモリ空間のうち、特に本発明に関連する部分を表すメモリマップであり、一例としてアドレス0~16Mの空間が見えている (左端の0 (H) ~FFFFF (H))。マスターCPU10は例えばアドレスA3-A4を指定すれば、ペリフェラル I/O を見ることができ、また例えばアドレスA1-A2を指定すれば、自分自身の予備側プログラムを見ることができる。結局、マスターCPU10は本発明による、ファームウェアの更新処理に必要な全ての情報を見ることができる。

【0029】図5においてペリフェラル I/O についてさらに具体的にアドレスの指定先を示している。図示するとおり全てレジスタである。これらレジスタに表示すべき情報は、前述した図3の内容そのものである。かくのごとく、マスターCPU10が有するメモリ空間の一部に定義されるペリフェラル I/O の領域に図1の更新処理手段40を実現する。ペリフェラル I/O 41は複数の更新処理レジスタ50を予め定義していて、これらの更新処理レジスタは、各スレーブCPU11、12毎に、プログラムメモリの現用面ACTおよび予備面STBYを個別に形成する第1バンク #1および第2バンク #2がいずれの面として機能しているかを表示する第1レジスタ51を含んでいる。

【0030】更新処理レジスタ50はまた、各スレーブCPU11、12毎に、データベースの現用面ACTおよび予備面STBYを個別に形成する第1バンク #1および第2バンク #2がいずれの面として機能しているかを表示する第2レジスタ52を含んでいる。更新処理レジスタ50はさらにまた、マスターCPU10がスレーブCPU11、12に対しリセットを指示する表示を行うために各スレーブCPU11、12毎に設けた第3レジスタ53を含み、この第3レジスタ53には、現用面ACTおよび予備面STBYの間の切替えが実行された

直後に当該スレーブCPUをリセットすべきことを指示するための表示を行う。

【0031】更新処理レジスタ50はまた、マスターCPU10がスレーブCPU11、12に対しリセットをかけてこれをリスタートさせたとき正常に立ち上がったか否かを表示するために各スレーブCPU毎に設けた第4レジスタ54を含んでいる。次にペリフェラル I/O 41を参照しながらマスターCPU10が実行する更新処理手順について述べる。

10 【0032】マスターCPU10がプログラムやデータベースのそれぞれについて新旧の切替えを行うときは、現在CPU11および12が実行しているサービスに影響を与えることのないよう、次に示すような手順を踏む。

<1> 不揮発領域のSTBY (予備面) 側に切り替えた後に、運用したいプログラムやデータベースをその領域にダウンロードする。そのとき、マスターCPU (CPU (M)) が自分自身のメモリマップ (図3、図4) 上にマッピングされている、スレーブCPU (CPU (S1) および/またはCPU (S2)) の不揮発領域

20 に対応するアドレスに対するデータの書き込みを行えば、バス (31) を介して接続されている不揮発メモリ (25および/または26) に当該データが格納される。

<2> バンクの切替えでACT (現用面) とSTBYとを反転させて運用面の切替え操作を行う。マスターCPUが自分自身のメモリマップ上にマッピングされている、スレーブCPUのバンク切替えレジスタ (51) を制御することによりこの操作が可能である。

30 <3> 切り替えられたプログラムやデータベースでソフトウェアを走らせるために、当該スレーブCPUをリセットし、リスタートさせる。この操作は、マスターCPUが自分自身のメモリマップ上にマッピングされている、当該スレーブCPUに対するCPUリセットレジスタ (53) を制御することにより行われる。

40 <4> マスターCPU (M) は、プログラムを切り替えた後のスレーブCPU (S1/S2) が正常に立ち上がったかを監視するために、CPU (M) のメモリマップ上にマッピングされている、スレーブCPUのCPUステータスレジスタ (54) をモニターし、異常があればスレーブCPUに対し直前に運用していた版のプログラムに切り戻すという操作を行うが、その手順は上記の切替え時の手順 (手順<2>および<3>) と同様である。

50 【0033】上記手順<1>~<4>が、ソフトウェアダウンロードからプログラムの版切替えに至る一連の手順であるが、ソフトウェアダウンロードおよび版切替えの処理において、マスターCPU (M) とスレーブCPU (S1/S2) との間でのCPU間通信 (13) による電文の送信および受信は行われず、したがって通信バス上での競合は起こらない。また、全ての操作がマスタ

一CPUのみによって行われ、切り替えられる側のスレーブCPUは当該操作に係る処理を行っていない。

【0034】本発明では上述したように、不揮発領域（メモリ）への書き込み、バンクの切替え（#1 ↔ #2）、CPUのリスタート、プログラム版の切り戻しを判断するための、スレーブCPUの状態監視等の各操作は全てマスターCPUが集中的に行っている。このことは、マスターCPU（M）側に処理が集約されること、つまりスレーブCPU（S1/S2）側は通常のアプリケーションに関する処理だけを行っていただければよいことを意味している。

【0035】そして従来のように、各スレーブCPUに、ソフトウェアダウンロードやプログラム版切替えの処理が始めから組み込まれていないため、当該処理を実施すべきソフトウェアに変更が生じたとしても、その変更は、マスターCPUのソフトウェアのみに対する変更により対処できる。このためスレーブCPU（S1/S2）のソフトウェアのうち、その変更に係るソフトウェアの変更ということはありませんので、アプリケーション処理（図1の“サービス1”や“サービス2”）に關しては、もともと使っていたROM内のソフトウェアをそのまま流用することができる。

【0036】このようにソフトウェアダウンロードやプログラム版の切替えの処理が全てマスターCPU（M）に集約されていることにより、スレーブCPU（S1/S2）側のソフトウェアが正常に動作していない状況下や、ソフトウェアが書き込まれていない状況下にあったとしても、ダウンロードや切替え処理は依然実行可能である。

【0037】また、スレーブCPU（S1/S2）への、プログラムやデータベースのダウンロードを、CPU間のシリアル通信（13）を介することなくマスターCPU（M）が直接行い、さらにその後の予備（STBY）から現用（ACT）への切替え制御もまた同様にCPU間のシリアル通信（13）を介することなくマスターCPUが直接制御を行うので、通信負荷やCPU負荷を増大させることは全くない。したがって、スレーブCPUが行う通常サービスの実行には全く干渉することがないから、これらスレーブCPUに不利をもたらすような影響を与えることはあり得ない。

【0038】次に実際の装置例に即して本発明を説明する。図6は本発明に係るシステムを、伝送装置を構成するユニット上に形成した実際の例を示す図である。本図の構成と、前述した図2および図3の構成とは実質的に同じである。したがって、図2および図3における対応部分には同一の参照番号または記号を付して示す。図6において新たに描かれた部分はEM1C・ユニット60とLCA61である。なお、図6中に記載された、EM1C、EM-S、EM-U、EM-L、LCAは実際の製品に用いるユニット名あるいはモジュール名であっ

て、本発明の理解のためには、これらの名称は直接関係ない。

【0039】EM1C・ユニット60は、本発明に係るマルチCPUシステムを構築する対象となる、一例としてのスイッチングネットワークにおける加入者系伝送装置の一部であり、カード状の基板（カード）である。実際には同様のカードが複数枚ボード上に差し込まれる。相互に隣接する複数のカードは、バス31に対しマルチドロップ形式で接続される。これら隣接のユニット（カード）も、各ファームウェアの更新に際しては図示するマスターCPU（EM-S）10によって集中的に制御される。

【0040】図中のLCA61は、EM1C・ユニット60内のハードウェア部分を統括的に制御する部分であり、LSIよりなる。また図中のPGM25、26は図2にも示したプログラムメモリ25、26であり、実例ではフラッシュメモリ（Flash）により構成される。前述した2面構成をなす第1バンク（#1）および第2バンク（#2）よりなる。

【0041】また図中のDBS27、28は図2にも示したデータベース27、28であり、実例では電氣的に消去可能なメモリ（EEPROM）により構成される。これもまた前述した2面構成をなす第1バンク（#1）および第2バンク（#2）よりなる。プロセッサからなるEM-U（S1）11およびEM-L（S2）12は、加入者サービスの監視制御機能、プロセッサEM-S（M）10はダウンロードおよび切替え機能をそれぞれ担う。前述のとおり、EM-S（M）10は、EM-U（S1）11、EM-L（S2）12の他に、隣接する複数のユニット（図示せず）上の各プロセッサに対するダウンロードおよび切替え制御も行う。

【0042】上述のとおり各プロセッサには、プログラム格納用のフラッシュメモリが2面ずつ実装され、それらはACTおよびSTBYのいずれか一方の役割を果たす。図7は図6に示すマスターCPU（EM-S）が有する実際のメモリマップを示す図（その1）、図8は同図（その2）である。

【0043】図7は第1スレーブデータベース（S1・DBS）27に関連するマッピング部分と、第1スレーブプログラムメモリ（S2・PGM）25に関連するマッピング部分を示している。図8はペリフェラルI/O41に関連するマッピング部分と、マスタープログラムメモリ（M・PGM）24に関連するマッピング部分を示す。ペリフェラルI/O41については同図中に詳細に展開して示し、図5の構成部分と対応する部分は同一の参照番号を付して示す。図8で新たに描かれたレジスタは、不揮発（NV）RAMステータスレジスタ71と、EEPROMステータスレジスタ72である。なお、図7および図8は本発明に関連するマッピング部分のみを抜粋して示す。

【0044】図7、8、9および10を参照して、EM-Lプロセッサ(S2)に対するプログラムのダウンロードおよび新旧版の切替えに関して、その手順を説明する。上述のとおり、図7および図8はダウンロードおよび切替え処理のためのマスターCPUとなるEM-S

(M)のメモリマップであって、プログラムの書き込みが行われるEM-L(S2)のフラッシュメモリ領域およびダウンロードおよび切替えの制御に必要なレジスタ群がマッピングされている。EM-S(M)と、EM-L(S2)のフラッシュメモリや各レジスタへの信号を出力するLCA61とは、直接バス31で接続されている。EM-L(S2)のフラッシュメモリは、EM-S(M)のメモリマップ上に、256Kバイトバウンダリでマッピングされる。EM-S(M)がペリフェラルI/O41上のレジスタ群を制御することによって、フラッシュメモリの全領域にアクセスすることができる。

【0045】図9は図6に示すマスターCPU(EM-S)により実行されるファームウェアの更新処理の実例を示すフローチャート(その1)、図10は同フローチャート(その2)であり、前述の図7および図8も参照しながら、ソフトウェアダウンロードおよび新旧版切替えのシーケンスを説明する。このシーケンスは(I)ソフトウェアダウンロード処理と、これに引き続く(II)新旧バージョン(版)切替処理に分けることができる。ただし、これら処理の対象としてEM-L(S2)を例にとり説明するが、EM-U(S1)についても同様である。

【0046】(I)ソフトウェアダウンロード処理
ステップS1(図9):プログラムの書き込みに際して、現在処理中のサービス(“サービス2”)に影響を与えぬように、プロセッサEM-L(S2)のフラッシュメモリのSTBY側にその新プログラムの書き込みを行う。そこでROMバンク制御レジスタ51'(図7および図8)を制御することにより、プロセッサEM-L(S2)のSTBY側フラッシュの先頭(図7のH)を、EM-S(M)のメモリマップ上にマッピングする。

【0047】すなわち、このステップS1は、<1>プロセッサEM-S(M)のメモリマップ上に、EM-L(S2)の書き込み対象のフラッシュエリアをマッピングするプロセスと、<2>マッピングされたそのフラッシュエリアへプログラムを書き込むプロセスとからなる。なお、上記<1>のプロセスは、図8に表すレジスタの制御による処理である。

【0048】さらに具体的には、図7において、
a)プロセッサEM-S(M)が、アドレスD0000(H)~D4000(H)に、256Kバイトのプログラムデータの書き込みを行い、

b)次の256Kバイトのプログラムデータの書き込みを行うために、上記ROMバンク制御レジスタ51'を制御し、プロセッサEM-L(S2)のフラッシュメモリ

の次の256Kバイトのエリアを、プロセッサEM-S(M)のメモリマップ上にマッピングする。

【0049】上記a)およびb)の操作は、プログラムデータの書き込みが終了するまで繰り返される。

(II)新旧バージョン(版)切替処理

ステップS2:プロセッサEM-L(S2)のフラッシュメモリにおけるACT側およびSTBY側に格納されているバージョン(版)のプログラムを、そのフラッシュメモリから読み出す。

【0050】さらに具体的には、切替え前に、プロセッサEM-L(S2)のフラッシュに書き込まれているプログラムのバージョンを読み出すために、ROMバンク制御レジスタ51'によりACT側のバージョンが書き込まれているフラッシュメモリのエリアを、プロセッサEM-S(M)のメモリマップ上にマッピングし、バージョンを読み出す。

【0051】次に上記と同様の手順でSTBY側のバージョンを読み出す。さらに詳細には、<1>プロセッサEM-S(M)のメモリマップ上に、プロセッサEM-L(S2)のACT側フラッシュメモリをマッピングし、<2>ACT側のプログラムのバージョン(版)を読み出し、<3>プロセッサEM-S(M)のメモリマップ上に、プロセッサEM-L(S2)のSTBY側フラッシュメモリをマッピングし、なお、上記<1>と<3>のプロセスは、図8に表すレジスタの制御による。<4>上記STBY側のプログラムのバージョン(版)を読み出す。

【0052】なお、上記のプロセス<1>および<3>は、図8に表すレジスタの制御による処理である。

ステップ3:プロセッサEM-L(S2)のSTBY側フラッシュメモリに書き込んだプログラムを有効にするために、図8に示すNVRAMステートレジスタ71により、フラッシュメモリのACTとSTBYを反転させ、新しいプログラムを書き込んだ面をACTとする。

【0053】ステップS4(図10):プロセッサEM-L(S2)をCPUリセットする。

ステップS5:このリセットによりリスタートしたプロセッサEM-L(S2)の障害やソフトランの状態を確認する。すなわち、図8のリセットレジスタ53により、プロセッサEM-L(S2)のCPUをリスタートさせ、新しいプログラムを動作させて、上記状況を確認する。

【0054】この状況確認は、図8のステータスレジスタ54により、プロセッサEM-L(S2)の障害やソフトランの状態をモニターし、異常なくプロセッサEM-L(S2)が新しいプログラムで立ち上がったか否かを監視することにより行う。この異常は、例えばウォッチドッグタイマにより行うことができる。

ステップS6:ステップS5での監視によって上記の異常が発見されれば、プロセッサEM-L(S2)のフラ

ッシュメモリのACTとSTBYを反転し、先に実行した切替えの切り戻しを行う。

【0055】すなわち、既述の更新処理手段40は、スレーブCPUをリスタートさせた後、このスレーブCPUが正常に立ち上がったか否かのステータスを監視して、正常に立ち上がっていないことが検知されたとき、現用面ACTと予備面STBYとの間の切替えを切り戻して元の状態を維持する。この場合、他方のプロセッサEM-U(S1)に対し既にプログラム切替えを実行していたときはこれも切り戻す必要がある。システム全体としてソフトウェアに不整合を生じてしまうからである。

【0056】ステップS7：上記の切り戻しを行ったときも、プロセッサEM-L(S2)に対し再びCPUリスタートをかけて初期化する。

【0057】

【発明の効果】以上説明したように本発明によれば、スレーブCPUのプログラムを格納するフラッシュメモリ、データベース(DBS)用のEEPROMおよび制御に必要なペリフェラルI/Oを、マスターCPUと直接バスで接続する構成をとることにより、各スレーブCPUにそれぞれ、プログラムおよびデータベース用データの書込みおよびプログラムの新旧切替え等のための制御機能を分散させることなしに、ソフトウェアダウンロードとプログラム切替えの各処理を全てマスターCPUに集約する。この結果、CPU資源の利用において通常サービスとの競合を避けることができると共に、スレーブCPU側のソフトウェアが正常に動作していなかったり、またソフトウェアが書き込まれていないような場合でも、ソフトウェアダウンロードやプログラム切替えの処理は最低限実行可能である。

【0058】また、スレーブCPUに対するプログラム切替え制御とプログラムおよびデータベース用データのダウンロードを、スレーブCPU間のシリアル通信を介さずに、マスターCPUが直接行うことにより一層効率的にそのダウンロードが実行でき、各スレーブCPUが実行中の通常サービスに何ら影響を与えることはない。

【0059】かくして、例えば伝送装置におけるソフトウェアダウンロードやプログラム切替え処理において、その処理の高効率化と、通常サービスへの影響の最小化に寄与するところが大きい。

【図面の簡単な説明】

【図1】本発明の基本構成を示す図である。

【図2】本発明に係るマルチCPUシステムの一実施例

を示す図(その1)である。

【図3】本発明に係るマルチCPUシステムの一実施例を示す図(その2)である。

【図4】マスターCPUが有するメモリ空間を表す図(その1)である。

【図5】マスターCPUが有するメモリ空間を表す図(その2)である。

【図6】本発明に係るシステムを、伝送装置を構成するユニット上に形成した実際の例を示す図である。

【図7】図6に示すマスターCPU(EM-S)が有する実際のメモリマップを示す図(その1)である。

【図8】図6に示すマスターCPU(EM-S)が有する実際のメモリマップを示す図(その2)である。

【図9】図6に示すマスターCPU(EM-S)により実行されるファームウェアの更新処理の実例を示すフローチャート(その1)である。

【図10】図6に示すマスターCPU(EM-S)により実行されるファームウェアの更新処理の実例を示すフローチャート(その2)である。

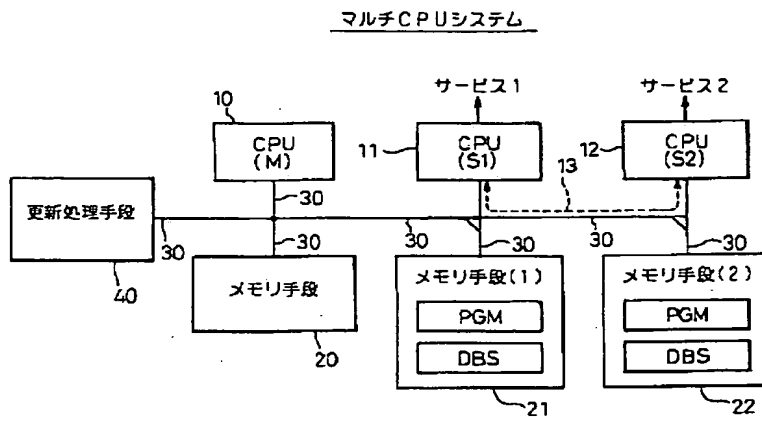
【図11】本発明が適用される好適な装置例を説明する図である。

【符号の説明】

- 10…マスターCPU(M)
- 11…スレーブCPU(S1)
- 12…スレーブCPU(S2)
- 13…CPU間通信
- 20, 21, 22…メモリ手段
- 24…マスタープログラムメモリ(M・PGM)
- 25…第1スレーブプログラムメモリ(S1・PGM)
- 26…第2スレーブプログラムメモリ(S2・PGM)
- 27…第1スレーブデータベース(S1・DBS)
- 28…第2スレーブデータベース(S2・DBS)
- 30…接続手段
- 31…バス
- 40…更新処理手段
- 41…ペリフェラルI/O
- 50…更新処理レジスタ
- 51…第1レジスタ
- 52…第2レジスタ
- 53…第3レジスタ
- 54…第4レジスタ
- PGM…プログラムメモリ
- DBS…データベース

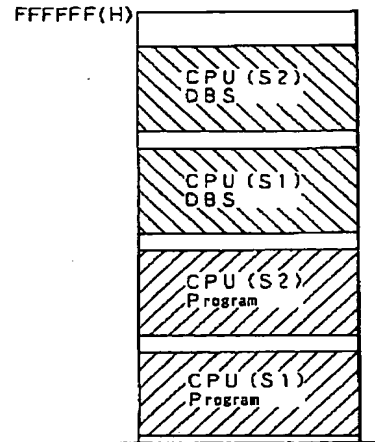
【図1】

本発明の基本構成を示す図



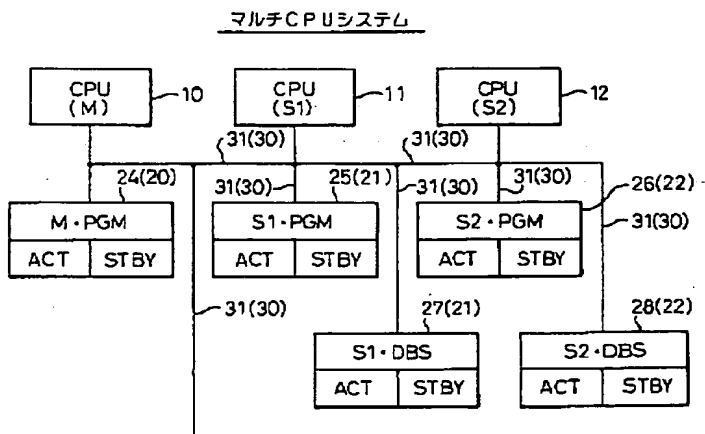
【図4】

マスターCPUが有するメモリ空間を表す図（その1）

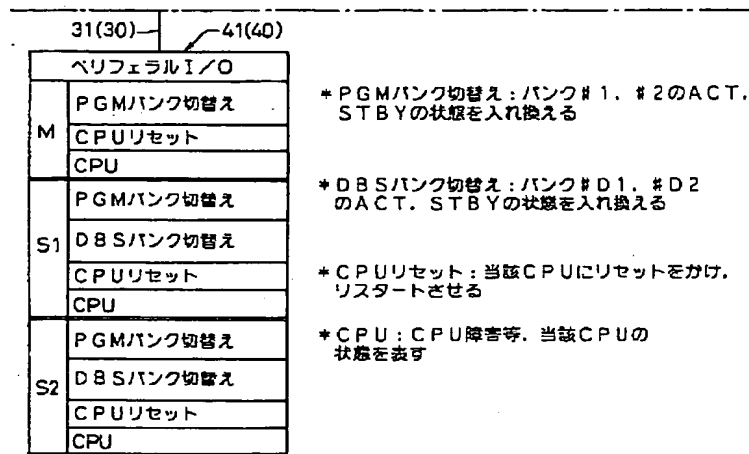


【図2】

本発明に係るマルチCPUシステムの一実施例を示す図（その1）

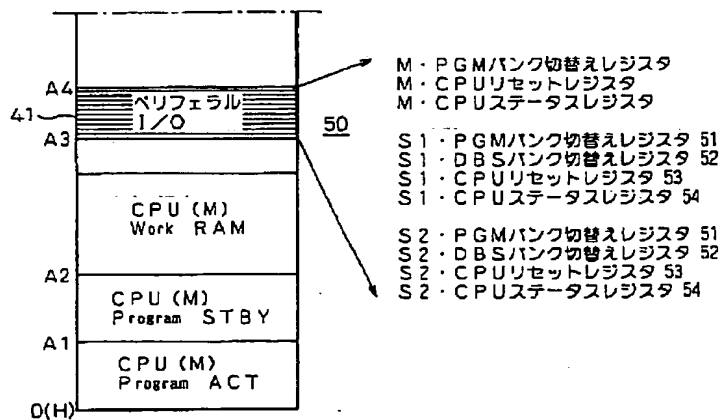


【図3】



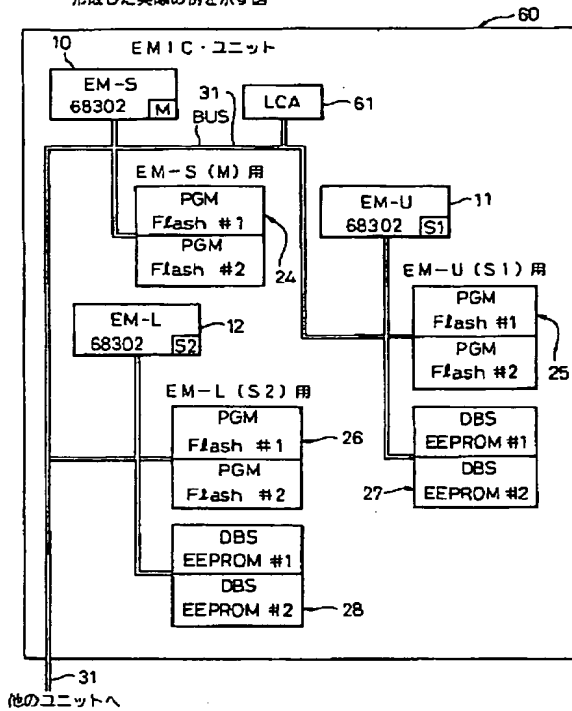
【図5】

マスターCPUが有するメモリ空間を表す図(その2)



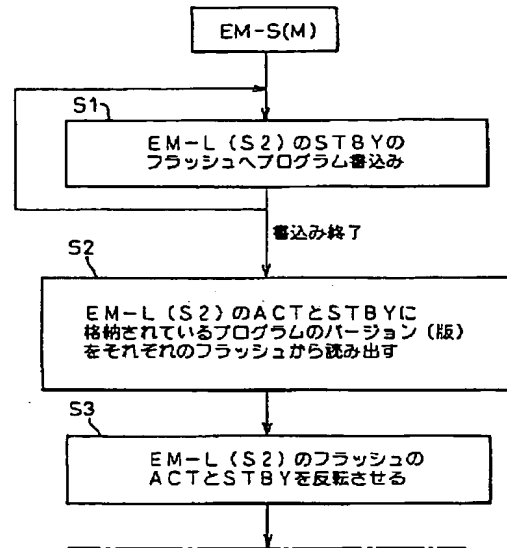
【図6】

本発明に係るシステムを、伝送装置を構成するユニット上に形成した実際の例を示す図



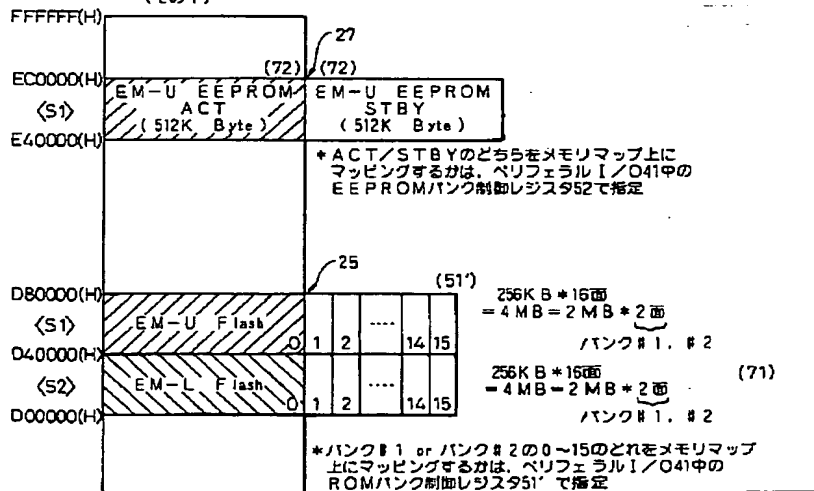
【図9】

図6に示すマスターCPU (EM-S) により実行されるファームウェアの更新処理の実例を示すフローチャート (その1)



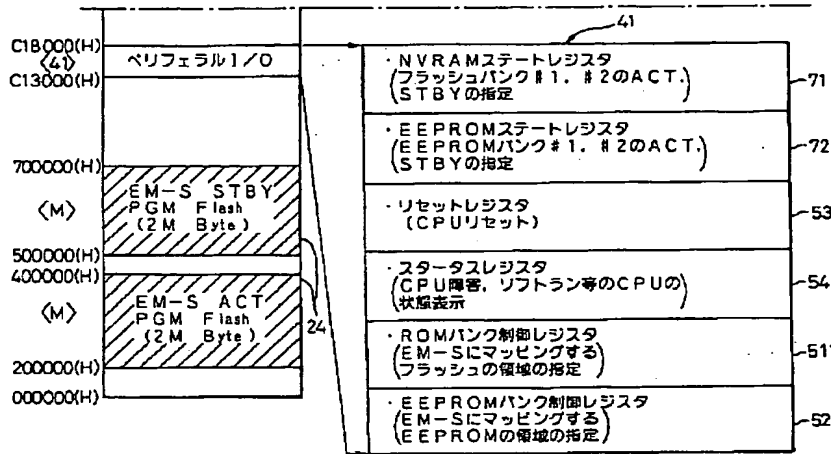
【図7】

図6に示すマスターCPU (EM-S) が有する実際のメモリマップを示す図 (その1)



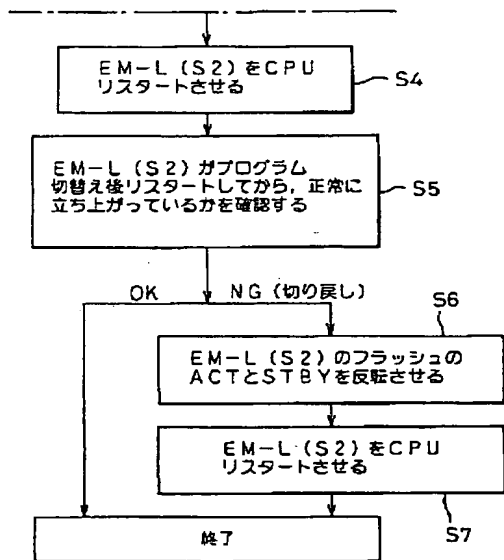
【図8】

図6に示すマスターCPU (EM-S) が有する実際のメモリマップを示す図
(その2)



【図10】

図6に示すマスターCPU (EM-S) により実行される
ファームウェアの更新処理の実例を示すフローチャート (その2)



【図11】

本発明が適用される好適な装置例を説明する図

